Post-Mortem
Assignment 09
April 2, 2018

We normally publish the post-mortem for an assignment after it has been marked and released. Here is a list of common errors provided by the graders for assignment 9.

General

• Many students are still missing design recipe components for locally defined helper functions. Remember that purposes and contracts are still required for any functions defined inside a local. However, purposes and contracts are not needed for locally defined constants.

• Some students emulated built-in functions to pass as arguments when using abstract list functions. This was mostly done by creating a lambda functions of the form (lambda (x) (f x)) or (lambda (x y) (f x y)), where f is any built-in function. To avoid overcomplicating code in this case by introducing an unnecessary lambda, simply pass the built-in function itself as an argument to a higher-order function.

• Based on the error messages, many students did not always ensure that the first argument given to build-list was a non-negative integer. This resulted in some correctness tests failing throughout the assignment.

Question 2

Generally well done.

Part a

• Some students did not correctly handle the cases where either of the strings could be empty, the case where the first string was longer than the second string, or the case where the two strings were of the same length.

• Some students did not include a test case where the two consumed strings were of the same length.

Part b

• Some students did not correctly handle the ‘count-unique case where every element from the list of strings appeared in the consumed string.

• Many students did not include a test case where the list of strings consumed by generate-flagger was empty.

• Some students did not correctly indicate in their contract that generate-flagger produces a function, or had formatting issues due to the produced function.

• Some students did not specify the requirement that the elements in the consumed list must be non-empty strings.